

PRAKTIKUM PEMROGRAMAN JARINGAN
JOBSHEET 4
“HTTP Request dan API”



DOSEN PENGAMPU:
Randi Proska Sandra, M.Sc.

OLEH:
Muhammad Alfarobi
23343011

PROGRAM STUDI INFORMATIKA
DEPARTEMEN TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI PADANG
2025

LATIHAN

A. HTTP Request & API

a. Mendapatkan API Access Key

1. Buatlah akun di <https://weatherstack.com/>. Bagian *company details* dapat dikosongkan.
2. Setelah itu silakan login dan anda akan masuk ke dashboard seperti gambar dibawah ini. Silakan perhatikan kode pada bagian “**Your API Secret key**” karena anda akan menggunakannya didalam program.



Thank you for signing up for APILayer!

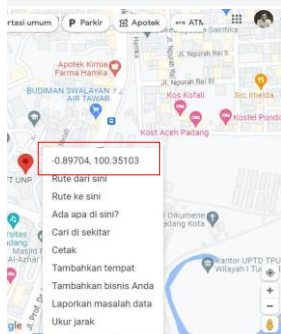
APILayer is your all-in-one platform for APIs.
Get a single API key from the APILayer Dashboard and unlock access to
hundreds of APIs without any setup.

VISIT DASHBOARD

Having trouble? Contact Us

3. Klik **3-Step Quickstart Guide** dan kemudian perhatikan base URL <http://api.weatherstack.com/> pada **Step 2: API Endpoints**. URL ini akan digunakan sebagai base URL untuk melakukan *request* ke API Weatherstack. Weatherstack menyediakan beberapa API Endpoints untuk mendapatkan data terkait cuaca diantaranya:
 - a) Current Weather: Mendapatkan data cuaca terbaru
 - b) Historical Weather: Mendapatkan data historis cuaca
 - c) Historical Time-Series: Mendapatkan data historis cuaca dalam rangkaian waktu tertentu.
 - d) Weather Forecast: Mendapatkan data ramalan cuaca hingga 14 hari.
 - e) Location Lookup: Mencari satu atau beberapa lokasi.
4. Bukalah google map dan carilah **Universitas Negeri Padang**. Klik kana dibagian map dan akan tampil informasi terkait latitude dan longitude. Informasi ini juga terdapat pada url google maps. Latitude dan longitude pada URL dan pada maps bisa saja berbeda tergantung dimana anda memposisikan klik kanan anda pada maps

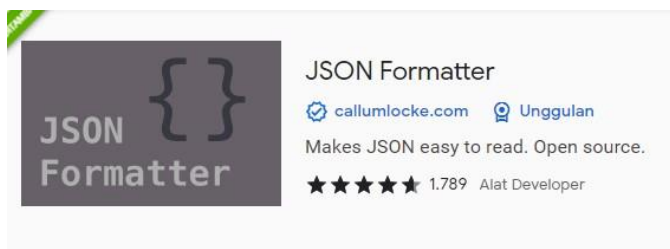
google.com/maps/place/Universitas+Negeri+Padang/@-0.8972206,100.3481587,17z/data=!4m6!3m...



5. Ketikkan URL berikut di tab baru browser anda
http://api.weatherstack.com/current?access_key=*****&query=*****
 Gantilah tandan bintang (*) berwarna **merah** dengan kode access API key anda, warna **hijau** dengan latitude dan warna **cokelat** dengan longitude. Pastikan tidak ada angka atau karakter yang tertinggal termasuk tanda minus (-) jika ada
6. Tekan enter dan akan tampil seperti gambar dibawah ini

```
{
  "request": {
    "type": "LatLon",
    "query": "Lat -0.90 and Lon 100.35",
    "language": "en",
    "unit": "m",
    "Sumatra",
    "lat": "-0.949",
    "lon": "100.354",
    "timezone_id": "Asia/Jakarta",
    "localtime": "2023-09-14H",
    "temperature": 26,
    "weather_code": 113,
    "weather_icons": [
      "https://cdn.worldweatheronline.com/
    ],
    "Sunny",
    "wind_speed": 5,
    "wind_degree": 212,
    "wind_dir": "SSW",
    "pressure": 1012,
    "precip": 0,
    "humid": 75
  }
}
```

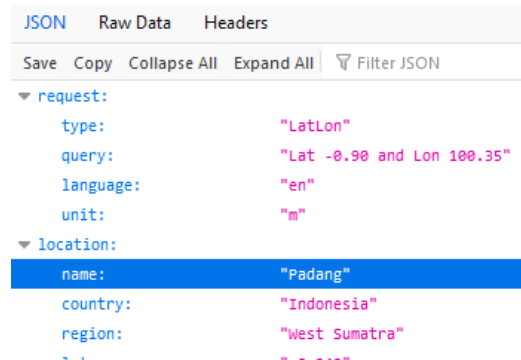
7. Tampilan diatas menggunakan format JSON. Agar tampilan menjadi lebih rapi, bukalah <https://chrome.google.com/webstore/category/extensions> kemudian cari JSON Formatters dan install ekstensi tersebut



8. Tampilan file JSON dari API Weatherstack akan menjadi seperti berikut ini

```
{
  "request": {
    "type": "LatLon",
    "query": "Lat -0.90 and Lon 100.35",
    "language": "en",
    "unit": "m"
  },
  "location": {
    "name": "Padang",
    "country": "Indonesia",
    "region": "West Sumatra",
    "lat": -0.949,
    "lon": 100.354,
    "timezone": "Asia/Jakarta",
    "localtime": "2023-09-14H",
    "temperature": 26,
    "weather_code": 113,
    "weather_icons": [
      "https://cdn.worldweatheronline.com/
    ],
    "Sunny",
    "wind_speed": 5,
    "wind_degree": 212,
    "wind_dir": "SSW",
    "pressure": 1012,
    "precip": 0,
    "humid": 75
  }
}
```

9. Jika anda menggunakan browser **Firefox**, maka **tidak perlu** menginstall extension atau add-on untuk tampilan seperti ini



10. Anda dapat menemukan informasi lengkap terkait cuaca di Universitas Negeri Padang melalui data yang ditampilkan seperti informasi lokasi, cuaca terkini, kapan cuaca di observasi, temperature, deskripsi dan lainnya. **PENTING!** Anda akan mengakses data ini melalui program

b. HTTP Request

1. Buatlah folder project baru pada visual studio code dengan nama **aplikasiCuaca**, lalu buatlah file javascript baru dalam folder tersebut dengan nama **app.js**.

2. Akses terminal melalui visual studio code, pastikan anda berada pada direktori folder yang baru anda buat. Lalu install **npm** dengan perintah **npm init** (sama seperti pada modul 2)
3. Setelah itu, silakan install module **postman-request** <https://www.npmjs.com/package/postman-request> dengan perintah **npm i postman-request**
4. Ketikkanlah kode berikut di file app.js

```
1. const request = require('postman-request')
2. const url =
  'http://api.weatherstack.com/current?access_key=*****
  &query=***** ,*****'
3. request({ url: url }, (error, response) => {
4. console.log(response)
5. // const data = JSON.parse(response.body)
6. // console.log(data)
7. // console.log(data.current)
8. // console.log(data.current.temperature)
9. })
```

5. Gantilah tanda bintang (*) seperti yang anda lakukan pada Langkah A no.5
6. Jalankan program tersebut melalui terminal dan pahami apa yang ditampilkan.
7. Jadikan kode pada baris ke-4 menjadi komentar dan kemudian **uncomment** baris ke-5 dan ke-6. Jalankan dan pahami apa yang ditampilkan
8. Jadikan kode pada baris ke-6 sebagai komentar, lalu **Uncomment** lagi kode pada baris ke 7 dan pahami apa yang ditampilkan
9. Jadikan kode pada baris ke-7 sebagai komentar, lalu **Uncomment** lagi kode pada baris ke-8 dan pahami apa yang ditampilkan.

```
● $ node app.js
{
  request: {
    type: 'LatLon',
    query: 'Lat -0.90 and Lon 100.35',
    language: 'en',
    unit: 'm'
  },
  location: {
    name: 'Padang',
    country: 'Indonesia',
    region: 'West Sumatra',
    lat: '-0.949',
    lon: '100.354',
    timezone_id: 'Asia/Jakarta',
    localtime: '2025-10-26 20:51',
    localtime_epoch: 1761511860,
    utc_offset: '7.0'
  },
}
```

```

current: {
  observation_time: '01:51 PM',
  temperature: 23,
  weather_code: 116,
  weather_icons: [
    'https://cdn.worldweatheronline.com/images/WSymbols01_png_64/WSymbol_0004_black_low_cloud.png'
  ],
  weather_descriptions: [ 'Partly Cloudy ' ],
  astro: {
    sunrise: '05:58 AM',
    sunset: '06:07 PM',
    moonrise: '09:26 AM',
    moonset: '09:55 PM',
    moon_phase: 'Waxing Crescent',
    moon_illumination: 17
  },
  air_quality: {
    co: '592.85',
    no2: '3.15',
    o3: '76',
    so2: '12.05',
    pm2_5: '13.35',
    pm10: '16.55',
    'us-epa-index': '1',
    'gb-defra-index': '1'
  },

```

```

    wind_speed: 4,
    wind_degree: 76,
    wind_dir: 'ENE',
    pressure: 1012,
    precip: 0,
    humidity: 81,
    cloudcover: 55,
    feelslike: 25,
    uv_index: 0,
    visibility: 10,
    is_day: 'no'
  }
}
{
  observation_time: '01:51 PM',
  temperature: 23,
  weather_code: 116,
  weather_icons: [
    'https://cdn.worldweatheronline.com/images/WSymbols01_png_64/WSymbol_0004_black_low_cloud.png'
  ],
  weather_descriptions: [ 'Partly Cloudy ' ],
  astro: {
    sunrise: '05:58 AM',
    sunset: '06:07 PM',
    moonrise: '09:26 AM',
    moonset: '09:55 PM',
    moon_phase: 'Waxing Crescent',
    moon_illumination: 17
  },

```

```

    air_quality: {
      co: '592.85',
      no2: '3.15',
      o3: '76',
      so2: '12.05',
      pm2_5: '13.35',
      pm10: '16.55',
      'us-epa-index': '1',
      'gb-defra-index': '1'
    },
    wind_speed: 4,
    wind_degree: 76,
    wind_dir: 'ENE',
    pressure: 1012,
    precip: 0,
    humidity: 81,
    cloudcover: 55,
    feelslike: 25,
    uv_index: 0,
    visibility: 10,
    is_day: 'no'
  }
}
23

```

10. Perlu diperhatikan pada const url bahwa **base url** untuk akses API menggunakan HTTP bukan HTTPS. Untuk menggunakan versi HTTPS maka dikenakan biaya. Pastikan ketika anda mengetikannya di browser juga menggunakan HTTP.
11. Perhatikan struktur file API dalam format JSON pada saat anda mengaksesnya melalui browser

```
{
  "request": {
    "type": "LatLon",
    "query": "Lat -0.90 and Lon 100.35",
    "language": "en",
    "unit": "m"
  },
  "location": {
    "name": "Padang",
    "country": "Indonesia",
    "region": "West Sumatra",
    "lat": "-0.949",
    "lon": "100.354",
    "timezone_id": "Asia/Jakarta",
    "localtime": "2025-10-26 20:51",
    "localtime_epoch": 1761511860,
    "utc_offset": "7.0"
  },
  "current": {
    "observation_time": "01:51 PM",
    "temperature": 23,
    "weather_code": 116,
    "weather_icons": [
      "https://cdn.worldweatheronline.com/images/wsymbols01_png_64/wsymbol_0004_black_low_cloud.png"
    ],
    "weather_descriptions": [
      "Partly Cloudy "
    ],
    "astro": {
      "sunrise": "05:58 AM",
      "sunset": "06:07 PM",
      "moonrise": "09:26 AM",
      "moonset": "09:55 PM",
      "moon_phase": "Waxing Crescent",
      "moon_illumination": 17
    },
    "air_quality": {
      "co": "592.85",
      "no2": "3.15",
      "o3": "76",
      "so2": "12.05",
      "pm2_5": "13.35",
      "pm10": "16.55",
      "us_aqi": "1"
    }
  }
}
```

12. Terlihat bahwa data tersebut memiliki hirarki yaitu request, location, current dimana didalamnya memiliki sub-hirarki atau data lagi. Kode pada no.4 baris ke-8 menunjukkan bahwa anda telah mengakses data -> current
-> temperature. Hal ini berarti anda juga dapat mengakses data lainnya. Sementara itu **response.body** menunjukkan bahwa anda mengakses keseluruhan data yang ditampilkan.

13. Buatlah file baru dengan nama **cekCuaca.js** dan ketikkan kode berikut ini

```
const request = require('postman-request')

const urlCuaca = 'http://api.weatherstack.com/current?access_key=*****&query=*****'

request({ url: urlCuaca, json: true }, (error, response) => {
  console.log('Saat ini suhu diluar mencapai '
    + response.body.current.temperature
    + ' derajat celcius. Kemungkinan terjadinya hujan adalah'
    + response.body.current.precip
    + '%')
})
```

14. **PENTING!** Jangan lupa mengganti tanda bintang dengan data anda masing-masing. Jalankan program tersebut dan pahami apa yang ditampilkan
15. Bukalah dokumentasi Weatherstack melalui link berikut ini <https://weatherstack.com/documentation> dan perhatikan bahwa Weatherstack memiliki beberapa opsi. Silakan klik **unit parameter**. Anda dapat melihat bahwa akan muncul units yang terkait dengan beberapa parameter suhu Celcius, Fahrenheit dan lainnya. Anda dapat menambahkan unit parameter ini keakhir pemanggilan API anda seperti berikut

```
const urlCuaca =
'http://api.weatherstack.com/current?access_key=*****&query=**
```

Options

Query Parameter

Units Parameter

Language Parameter

JSONP Callbacks

Units Parameter

Available on: All plans

By default, the API will return all results in metric units. Aside from metric units, other common unit formats are supported as well. You can use the `units` parameter to switch between the different unit formats Metric, Scientific and Fahrenheit.

For Metric:

Parameter	Units
<code>units = m</code>	temperature: Celsius
<code>units = m</code>	Wind Speed/Visibility: Kilometers/Hour
<code>units = m</code>	Pressure: MB - Millibar
<code>units = m</code>	Precip: MM - Millimeters
<code>units = m</code>	Total Snow: CM - Centimeters

16. Gantilah huruf **m** dengan parameter lainnya yang tersedia pada dokumentasi weatherstack. Silakan jelajahi dokumentasi tersebut anda akan menemukan juga parameter *language* (bahasa) dan lainnya.

c. Latihan 1 – API Access Weatherstack

1. Pada kode yang ada pada Langkah b no.13, tambahkan kode agar anda bisa mengakses `weather_descriptions` (deskripsi cuaca)
2. Tampilkan teks berikut sesuai dengan keinginan anda.
3. Perlu diperhatikan bahwa data `weather_descriptions` menggunakan format array sehingga anda perlu menambah `[]` di akhir pemanggilan data. Misal: seandainya `temperature` dalam format array maka akan menjadi `response.body.current.temperature[0]`

```
const request = require('postman-request')

const urlCuaca =
'http://api.weatherstack.com/current?access_key=ad3f07c5c5973ebae43e7f47368d68d2&query=-0.8976644584004788, 100.3493652562926'
```

```
// request({ url: urlCuaca, json: true }, (error, response) => {
//   console.log('Saat ini suhu diluar mencapai '
//     + response.body.current.temperature
//     + ' derajat celcius. Kemungkinan terjadinya hujan adalah'
//     + response.body.current.precip
//     + '%')
// })
// })
```

```
request({ url: urlCuaca, json: true }, (error, response) => {
  if (error) {
    console.log('Tidak dapat terkoneksi ke layanan cuaca.')
  } else if (response.body.error) {
    console.log('Tidak dapat menemukan lokasi.')
  } else {
    console.log(
      'Saat ini suhu di luar mencapai '
      + response.body.current.temperature
      + '°C. '
      + 'Kemungkinan terjadinya hujan adalah '
      + response.body.current.precip
      + '%. '
      + 'Kondisi cuaca saat ini: '
      + response.body.current.weather_descriptions[0] + '.'
    )
  }
})
```

```
$ node cekCuaca.js
```

```
Saat ini suhu di luar mencapai 23°C. Kemungkinan terjadinya hujan adalah 0%. Kondisi cuaca saat ini: Partly Cloudy .
```

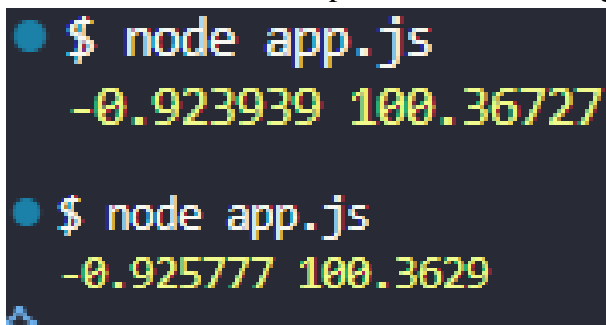
d. Latihan 2 – API Mapbox

1. Silakan buat akun <https://www.mapbox.com/>. Setelah itu login dan cek menu tokens.
2. Bukalah dokumentasi API mapbox melalui link berikut ini <https://docs.mapbox.com/api/search/geocoding/> Carilah **Forward Geocoding** (menu dibagian kanan)
3. Pahami dokumentasinya dan lihat bagaimana cara pemanggilan API nya. Perhatikan data yang ada pada tabel dan juga contoh pada **example request**
4. Berikut adalah salah satu example request yang mereka sediakan https://api.mapbox.com/geocoding/v5/mapbox.places/Washington.json?limit=2&access_token=pk.eyJ1IjoicHJvc2thOTkiLCJhIjoieY2xsbmJkYWppMDBIYzNyYXcwNmM3p6OSJ9.9S7iRjDXb1JqUbpKK949ew
5. Gaantilah teks warna **hijau** dengan kota atau tempat yang ingin anda cari, misal: **Padang** atau **Universitas Negeri Padang**. Lalu, ganti teks warna **merah** dengan token anda.
6. Cobalah akses di browser dan lihat data yang ditampilkan
7. Lalu cobalah lakukan **request data** dengan menambahkan kode berikut ke file

app.js

```
const geocodeURL =  
'https://api.mapbox.com/geocoding/v5/mapbox.places/*****  
.json?access_token=*****&limit=1'  
  
request({ url: geocodeURL, json: true }, (error,  
response) => {  
  const latitude = response.body.features[0].center[1]  
  const longitude =  
response.body.features[0].center[0]  
  console.log(latitude, longitude)  
})
```

8. Gantilah teks warna **hijau** dengan lokasi atau tempat yang ingin anda cari dan teks warna **merah** dengan token API Mapbox anda
9. Limit=1 digunakan untuk membatasi pencarian data agar hanya menampilkan 1 pencarian. Cobalah ganti angka 1 menjadi 2 atau 3, lalu gantilah array [0] pada features dengan angka 1 atau 2. Lakukan untuk kedua variable latitude dan longitude
10. Perhatikanlah bahwa tampilan latitude dan longitude pada terminal telah berganti

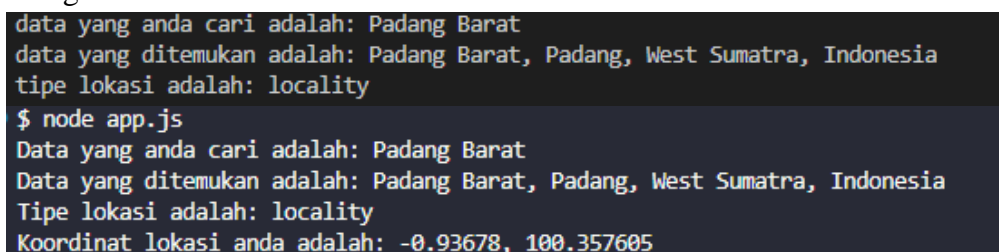


```
$ node app.js  
-0.923939 100.36727  
  
$ node app.js  
-0.925777 100.3629
```

Penting: output yang tampil tergantung pada lokasi yang anda acari. Meski limit anda tingkatkan menjadi lebih dari 1, data yang didapatkan tidak selalu lebih dari 1 karena bergantung kepada jumlah data yang disediakan oleh Mapbox API. Cobalah mencari lokasi yang populer seperti Jakarta atau Bali.

e. Latihan 3 – memanggil data API

1. Lanjutkanlah program pada **bagian d**. Lakukanlah pemanggilan data sebagaimana yang telah anda pelajari pada saat mengakses API weatherstack untuk menampilkan query, place_name dan place_type. Contoh tampilannya adalah sebagai berikut



```
data yang anda cari adalah: Padang Barat  
data yang ditemukan adalah: Padang Barat, Padang, West Sumatra, Indonesia  
tipe lokasi adalah: locality  
$ node app.js  
Data yang anda cari adalah: Padang Barat  
Data yang ditemukan adalah: Padang Barat, Padang, West Sumatra, Indonesia  
Tipe lokasi adalah: locality  
Koordinat lokasi anda adalah: -0.93678, 100.357605
```

2. Lakukanlah pemanggilan data dari API weatherstack dan mapbox API hingga menampilkan output seperti berikut. **Penting!** Data lokasi bisa berbeda tergantung lokasi yang anda cari. Pastikan data lokasi yang anda cari dengan API mapbox

sesuai dengan koordinat yang anda berikan pada API weatherstack

```
Koordinat lokasi anda adalah -0.924759, 100.363256
Data yang anda cari adalah: padang
Data yang ditemukan adalah: Padang Barat, Padang, West Sumatra, Indonesia
Tipe lokasi adalah: locality
Saat ini suhu di Padang mencapai 26 derajat celcius.
Kemungkinan terjadinya hujan adalah 0%

$ node app.js
Koordinat lokasi anda adalah -0.93678, 100.357605
Data yang anda cari adalah: Padang Barat
Data yang ditemukan adalah: Padang Barat, Padang, West Sumatra, Indonesia
Tipe lokasi adalah: locality
Saat ini suhu di Padang Barat mencapai 23 derajat celcius.
Kemungkinan terjadinya hujan adalah 0%
Deskripsi cuaca: Cloudy
```